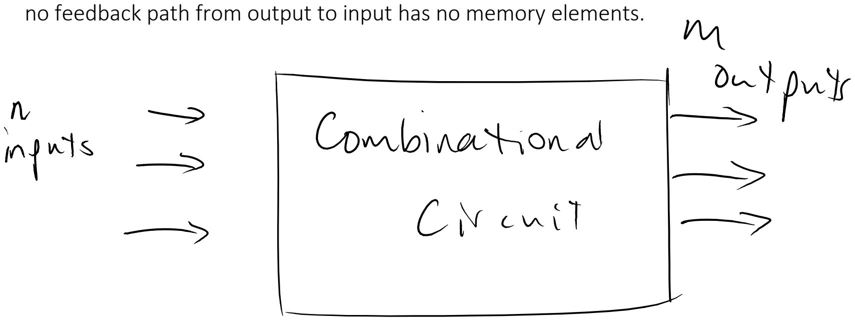
Combinational Logic

Monday, October 13, 2014 12:33 PM

Combinational logic consists of input variables, logic gates, and output variables. CL is a logic circuit that has no feedback path from output to input has no memory elements.



Analyzing Combinational Logic Circuits

- Obtain a function for each output
- Derive a truth table for each output
- Interpret the operation

Code Conversion: BCD to Excess-3

- *Determine the inputs and the outputs.*
- *Derive a truth table for each circuit.*
- *Obtain a function for each output.*
- *Design the circuit.*

Half Adder is a very basic addition circuit which can sum up two bits.

Designing a Half Adder

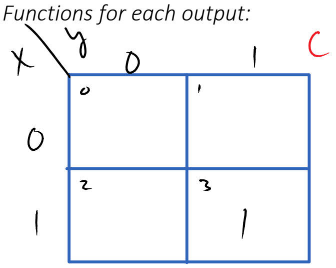
- *Inputs and Outputs:* Two input bits X and Y. We also need *two* output bits because if X and Y are both 1, then the sum will be 10 (base 2).
- *Truth Table*

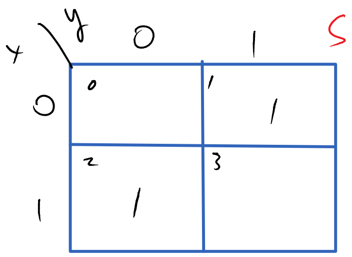
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

C = Carry

S = Sum

- *Functions for each output:*

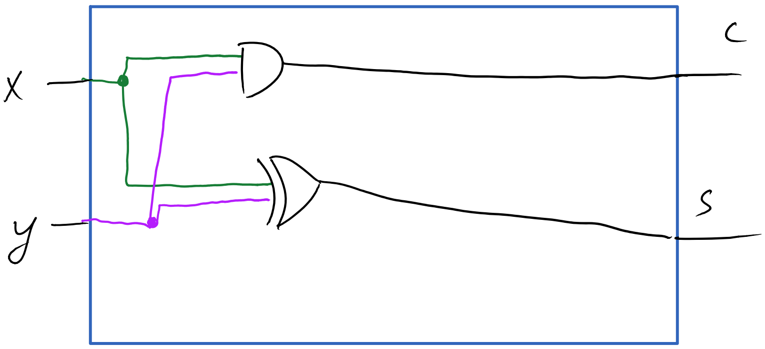




$C = xy$

$S = x'y + xy'$

- *Design for the circuits:*



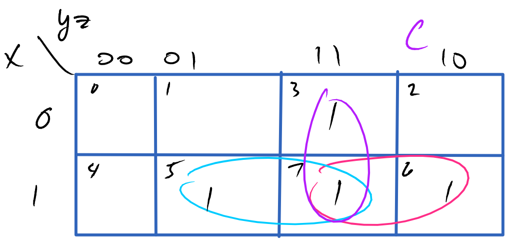
Full Adder will add *three* bits.

Inputs: 3
Outputs: 2 (Biggest output = 11)

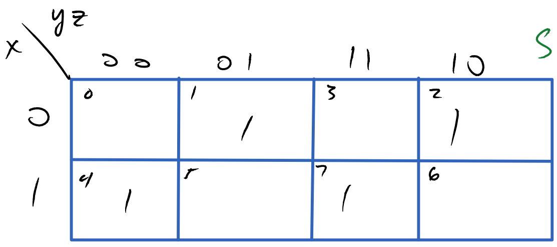
Truth Table

x	y	z	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Function Mapping



x \ yz	00	01	11	10
0	0	1	3 1	2
1	4	5 1	7 1	6 1

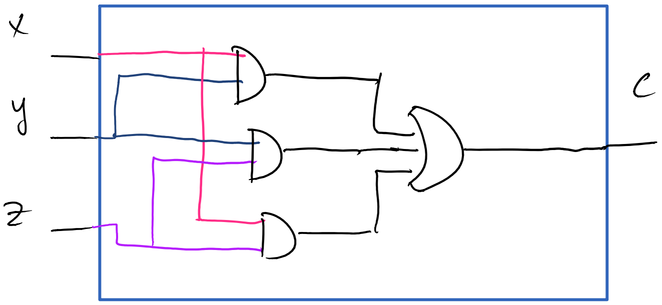


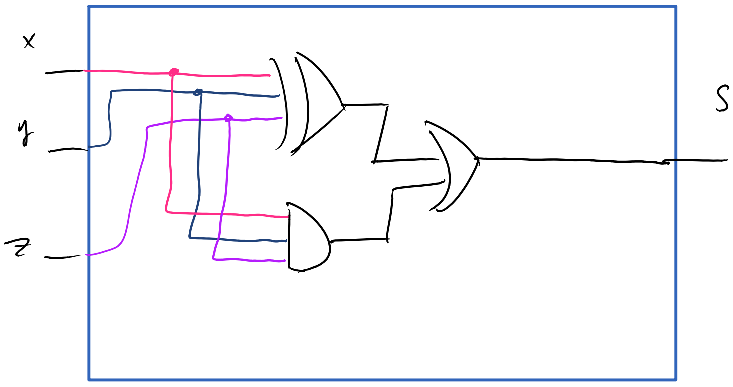
x \ yz	00	01	11	10
0	0	1 1	3	2 1
1	4 1	5	7 1	6

$$C = yz + xz + xy$$

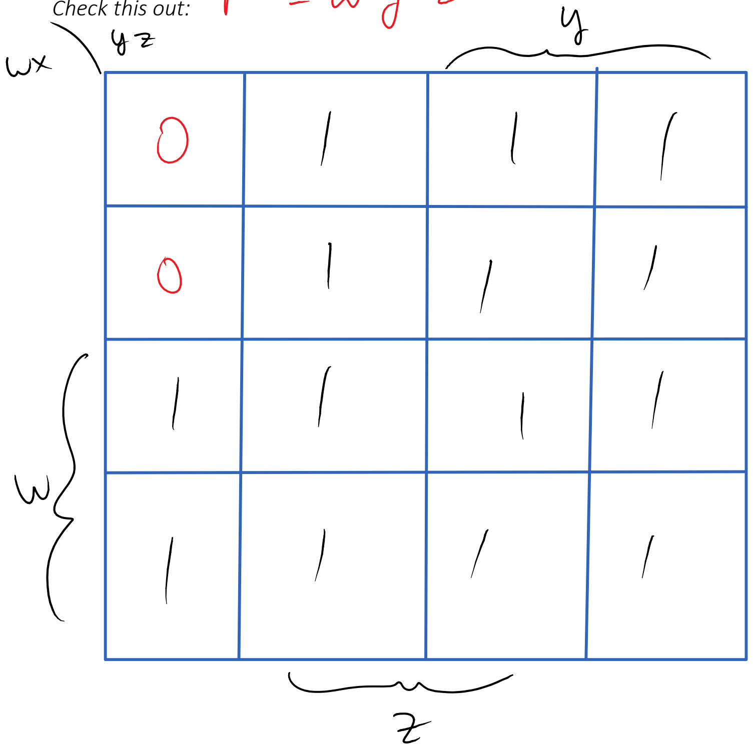
$$S = xy'z' + x'y'z + xyz + x'yz'$$
$$= (x \oplus y \oplus z) + xyz$$

Circuit Design





Check this out: $F' = w'y'z'$



0	1	1	1
0	1	1	1
1	1	1	1
1	1	1	1

$$F = (F')'$$

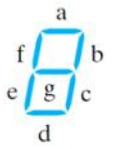
$$= w + y + z$$

What I just did:

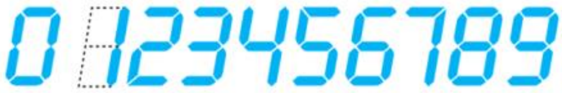
I was given a K-map with a very large number of minterms. Instead of grouping minterms together to derive my function F, I grouped the maxterms to get F'. Then, because F'' = F, I applied de Morgan's law and determined the simplest possible representation of function F.

Wow!

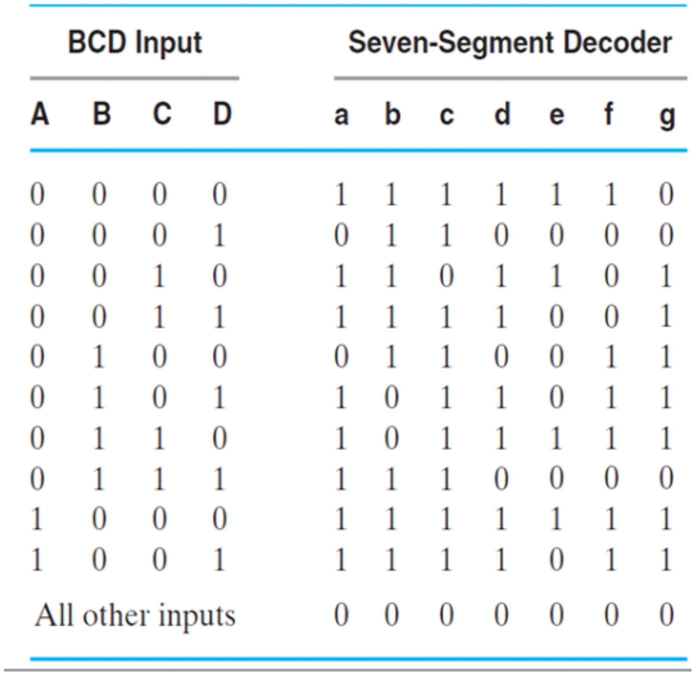
Build a circuit to display a decimal digit (BCD) in a 7-segment component.



(a) Segment designation



(b) Numeric designation for display



BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0